SURROUND AI SQUAD 1

AI Literacy Tool

project handover document T2/2020

Project Client

The tribe leader for this project is Chandan Karmakar.

Academic Mentor/Supervisor

The academic mentors for this project are Jonathan Kua and Abbas Kudrati.

Project Team [Data Intelligence Consulting – PG – Surround AI Squad 1 T2 2020]

|  |  |  |
| --- | --- | --- |
| Student Id | Student Name | Role |
| Student#1 id | Aamir Mapkar | Team Leader and Front-end developer |
| Student#2 id | Sameer Mediratta | Front-end developer |
| Student#3 id | Alwin Antony | Back-end developer |
| Student#4 id | Osama Khalid | Back-end developer |
| Student#5 id | Shanya Vairawanathan | Front-end developer |
| Student#6 id | Yinghui Liu | Back-end developer |
| Student#7 id | Prafful Theja Gadadasu | Back-end developer |

# Project Overview

# Please provide a description of what the project is about and list the overall aims and deliverables of the project.

# Document Purpose

This document records the transfer of all the information and artefacts produced during the duration of this Trimester (Trimester 1, 2020). The handover document and delivery package includes the following:

1. Please specify what is included in your delivery and documented here. This may include initial proposal, plans, status reports, demos, iteration reports, squad reflections, source code – if any, executables – if any, key open issues, assumptions, vision for future work on the project, or other relevant documents. Please make sure to refer to document name or folder name as needed.
2. ..
3. …

# Completed Deliverables

**Product Features:** We have developed the product in three different iterations where every feature is explained.

1. **Iteration 0:** We Modified old website to new website using Python framework in which

Back-End Developing Tools are:

* 1. Django for Server-Side web-framework
  2. OpenCV (Open Computer Vision)
  3. Neural Networks-Machine Learning (DNN, CNN)
  4. Python Libraries like NumPy, argparse.
  5. Natural Language Toolkit (nltk)
  6. Python Modules like pickle, caffemodel, dlib.

Front-End Developing Tools are:

1. Bootstrap
2. Django (Static Libraries)
3. Cascading Style Sheet (CSS)
4. Hypertext Mark-up Language (HTML)
5. JavaScript

For New Website, we have initialized four working html pages

* Index.html as Home Page
* Examples.html

Which consists of Three AI examples in which two Basic AI examples are completed for this semester.

* Age\_gender\_example.html as Age and Gender Detection Page
* Sentiment\_analysis\_example.html as Sentimental Analysis Page

In this Iteration, we modified the Home page with different UI modules like

* Hovering effect and on-mouse effects
* Useful Information and content related to Artificial Intelligence of what is data and how it works? followed by how machine works using Tree-type structure of 7 easy steps including Demo Video about Importance of AI and usage of it in modern world.

We also created Examples.html page for future iterations to create and develop two basic AI examples in it.

**Upskilling**: Every squad member upskilled the concepts of the Front-End concepts like HTML, JavaScript, CSS, Django for development of this Iteration.

1. **Iteration 1:** In this Iteration, we have developed the Age\_gender\_example.html page

For creating Age and Gender Detection, we required the best possible User interface along with Back-end development on Detecting Age and Gender.

* As the purpose of this website is to interact the children with the concepts of AI, we have created a basic interface of an Artificial-Intelligent robot called DASH.

A close up of a sign

Description automatically generated

* Dash is a robot where initially it knows nothing and interacts with the user to learn the Artificial-Intelligent concepts like detecting the age and gender of the person.
* We have created a Drag and drop method which makes user more user-friendly. We have developed four different module blocks, where each block has to drop on-to the robot for each function to be performed.
* The four Back-End Functions are:
  + 1. Detecting Face:

For Detecting the Face, we have used OpenCV and convolutional neural networks (CNN) concepts.

We have also used open source of caffemodel and deploy.prototxt CAFFE(Convolutional Architecture for Fast Feature Embedding) framework.

The Output will be displayed on the shape predictor with Rectangular box where the face is located on the given image

* + 1. Detecting the Features of the Face

We have used dlib face recognition toolkit for detecting the facial landmarks and assigned them to the shape predictor for the specific shape we required around the face of the image which we give, in our case we have used red-rectangle shape around the face.

The Output will be displayed on the facial predictor with Rectangular box where the face is located on the given image.

* + 1. Detecting the Age

We divided the age detection into two tasks in which first task was to detect the age of range 7 and second task was to detect the age of range 9.

For Detecting the Age, we have used OpenCV and convolutional neural networks (CNN) concepts.

We have also used open source of age\_net.caffemodel and Age\_deploy.prototxt CAFFE(Convolutional Architecture for Fast Feature Embedding) framework.

The Output of the Age will be displayed on the shape predictor with appropriate range of age.

* + 1. Detecting the Gender

For Gender-detection also, we have used the OpenCV and convolutional neural networks (CNN) concepts.

We have also used open source of gender\_net.caffemodel and gender\_deploy.prototxt CAFFE(Convolutional Architecture for Fast Feature Embedding) framework.

FaceNet is the variable from gender\_deploy.prototxt where it detects the person either Male or Female and displays onto the image as an output.

The Output of the Gender will be displayed on the shape predictor of either Male or Female.

* + - * Dash learns from all these functional blocks whenever user drag and drops on to it.
      * After Each functional Block is drag onto the robot, it will show the output next to it.

**Upskilling:** Front-End Team upskilled the concepts of HTML, JavaScript, CSS, Django for development of this Iteration. Back-end Team upskilled the concepts of OpenCV (Open Computer Vision), Neural Networks-Machine Learning (DNN, CNN), Python Modules like pickle, caffemodel, dlib.

**Worked on Additional-Feature:** We have also creating additional features like

* + - Functional Blocks of Face detection and Features of the face has to be in order such a way that Face Detection has to be first to learn by the robot, otherwise the features or Age and Gender won’t allow DASH to learn the Age and Gender functions.
    - While the Dash knows the Face and Features of the image, it can be any order for the detection of Age or Gender.

1. **Iteration 2:** In this Iteration,
   1. We have developed the Sentiment\_analysis\_example.html page
   2. Fixed different kinds of minor bugs in iteration 1
   3. Included the side navigation bar in both example pages to switch quickly and easier into other example.
   * Sentiment analysis is a common NLP task, which involves classifying texts or parts of texts into a pre-defined sentiment. We have used the Natural Language Toolkit (NLTK), a commonly used NLP library in Python, to analyze textual data.
   * For Sentiment Analysis, we have provided the user with two options, choose a random text message, or create his/her own custom message to find the sentiment.
   * **For Random message**, we have created a database and stored some random message from some random twitter account and input some random tweets from the database.
   * **For Custom message,** User can type some random text message he would like to find the sentiment of it.
2. For creating Sentimental Analysis page, we used the same type of User interface which we used in Age and Gender detection example, so that the user will find both examples easy to interact.
   * + - We similarly created a Drag and drop method as same in age and gender detection page which makes user more user-friendly. We have developed four different module blocks, where each block has to drop on-to the robot for each function to be performed.
       - The four Back-End Functions are:
         1. **Tokenization**:

* Splitting strings into smaller parts called tokens.
* We have used the python library tool NLTK variable called nltk.word\_tokenize where it will cut the sentence and store it in the form of array whenever it see space in between the words.
* For Example: “This project name is AI Literacy Tool.”

The output will be like “This”, “project”, “name”, “is”, “AI”, “Literacy Tool.” Stored in Array format and we have change them into string format to make it easy for further steps.

* + - * 1. **Stemming**:
* The process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language
* We have used the pre-defined function called lemmatizer as WordNetLemmatizer() from library for the stemming process to the given input text.
* For example: It will reduce the words like “chocolates”, “chocolatey”, “choco” to the root word, “chocolate”.
  + - * 1. **Stop** **Word** **removal**:
* Here, we have removed the words which does not give any meaning to the sentence or information to data.
* We have removed three different types of stop-word formats, they are:

Hyperlinks

Twitter handles like @ after their names as we have used twitter data.

Punctuation and special characters like | [$-\_@.&+#]|[!\*\(\),] are removed in this step.

* We have also used the build-in set of stop-words in NLTK which we have downloaded separately in stop\_words\_removed.ipynb file using nltk.download(‘stopwords’)
  + - * 1. **Sentiment** **analysis**:
* Finally, in this step, we have created a classifier to train the set of data to be either positive or negative.
* We created custom variables called positive\_tweet\_tokens and negative\_tweet\_tokens which are tokenized, stemmed and removed stop words from it.
* We have used NaiveBayesClassifier imported from NLTK library for accuracy.
* Classifier will train the data to be positive if the data has information features like 😊, welcome, glad, Good. Etc.

Classifier will train the data to negative if the data has information features like ☹, sad, bad. Etc.

* The output will be displayed as “Happy sentiment” if the statement is positive and “Unhappy sentiment” if the statement is negative.
* Dash learns from all these functional blocks whenever user drag and drops on to it.
* After Each functional Block is drag onto the robot, it will show the output next to it.

**Upskilling:** Front-End Team upskilled the concepts of HTML, JavaScript, CSS, Django for development of this Iteration. Back-end Team upskilled the concepts of OpenCV (Open Computer Vision), Neural Networks-Machine Learning (DNN, CNN), Python Modules like pickle, caffemodel, dlib.

1. Fixed the bugs in Iteration 1 like,

* Input Image pixel modifying: Any size of the image can be modified and reduced the pixel quantity so that it can fit exactly as required in the given input box.
* Resolved and modified the “User Interface Design layer collision”
* Saving Images into local database.
* Resolved the Django webserver in different types of devices.

1. Included the side navigation bar in both example pages to switch quickly and easier into other example.

**Upskilling Hours by the respective Squad Members:**

Upskilling Hours

|  |  |  |
| --- | --- | --- |
| **Team Member​** | **Junior/Senior​** | **Up-skilling Hours​** |
| Aamir Mapkar​ | Senior​ | 16​ |
| Alwin Antony​ | Senior​ | 6.3​ |
| Eavan Liu​ | Senior​ | 13​ |
| Osama Khalid​ | Senior​ | 8​ |
| Sameer Mediratta​ | Senior​ | 3.3​ |
| Shanya Vairawanathan​ | Junior​ | 26.5​ |
| Prafful Gadadasu​ | Junior​ | 10​ |
| Shengda Yuan​ | Junior​ | 0\*​ |

**Squad Members primarily responsible for completion of their tasks for their respective deliverables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task​** | **Status​** | **Comments​** | **Members​** |
| Implement improvements and bug-fixes. ​ | 100%​ | Implemented and working as expected. ​ | Aamir, Sameer. ​ |
| Generate backend scripts forperforming sentiment analysis. ​ | 100%​ | All scripts implementedand working as expected. ​ | Alwin,Prafful,Eavan,Osama.​ |
| Generate front-end page andintegrate it with backend. ​ | 100%​ | Backend scriptsintegrated with front-end and working as expected. ​ | Aamir, Sameer, Alwin,Shanya.​ |
| Create a design for side bar. ​ | 100%​ | Implemented and designflows created. ​ | Shanya. ​ |
| Implement the sidebar with created design. ​ | 100%​ | Implemented and working as expected. ​ | Shanya, Aamir. ​ |

# Planned work

Planned features for the future:

1. Improve the existing examples by changing the complexity based on the user’s experience level i.e. Beginner, Intermediate, Advanced.
2. Improve the layout of the website and create walkthroughs to enable new users to easily navigate through the website. Since the website is intended for kids, having a walkthrough would make it easier for them interact and learn.
3. Create more examples that explain a specific AI/ML example in a similar way as the previous example i.e. using a drag and drop functionality.

It is important to note that these features are tentative and deliverables for the next semester should be decided after consultation with the product owner.

There are no features that we started and did not finish implementing.

# Open issues

Some of the prominent issues we faced were as follows:

* 1. We have not been able to deploy the website due to issues with static location of CSS and JS. This is mostly due to limited knowledge of Django’s internal working and can be potentially resolved by upskilling on it a bit more.
  2. There are multiple CSS and JS files and they can potentially be combined to make imports and code look cleaner.

Apart from these, there are no major issues that need to be fixed immediately in the project.

# Lessons Learned

Some important takeaways for us are as follows:

1. Django, although promises to make web development smooth, is very difficult to understand within a short span of time to be used effectively. Creating a cheat sheet to help new users get started with it would potentially cut down the upskilling time required. Changing the backend to any other service could be an uphill task without sound knowledge of Django.
2. Bootstrap was an excellent choice for front-end development as it enabled us to cut down the time spent on website development and focus more on the implementing the examples which is the core functionality of the website.
3. Implementations of the ML examples can be looked into to further optimize them since at the moment they are using varying libraries with pre-trained models which take a lot of size, leading to a large increase in the project storage size.

# High-level architecture of the product

Diagram

Description automatically generated

# Source code

Please add all necessary details for your project’s source code – links, key components, classes, database components, URLs of online hosted repositories, etc. Please make sure to include your project’s source code in the delivery package if it is hosted on a server outside Deakin.

# User manual

Please provide detailed instructions on how end user can use your system (where applicable).

# Other relevant documents

Please provide any relevant information not covered in the above sections.

# Appendices

Appendix A:

# **Bitbucket Repository:** Student Bitbucket repository for developing and modifying by the team members is provided in the link below [Click to view](https://bitbucket-students.deakin.edu.au/projects/D2IC-PG/repos/surround-ai-squad-1_2020t2)

Appendix B:

* **Worklog:** Working hours spend by each squad member to the project is provided by the link [Click to view.](https://teams.microsoft.com/l/file/AD0F9F09-7271-4B6D-B5C7-8485A72AC96C?tenantId=d02378ec-1688-46d5-8540-1c28b5f470f6&fileType=xlsx&objectUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting%2FShared%20Documents%2FSurround%20AI%2FT2%202020%20(Squad%201)%2FWorkbook%202020%20T2.xlsx&baseUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting&serviceName=teams&threadId=19:7e7092a286674cf99464a987cc2694bd@thread.tacv2&groupId=10567a03-e6be-45e6-b917-f7b8920e9c9b)

Appendix C:

* **Trello Board**: Trello Board activities, product back-logs and completed tasks for different iteration is provided in the link [Click to view](https://trello.com/b/ALyogRh7/surroundait22020)​

Appendix D:

* **Minutes of Meeting:** Hours spend for squad session, stand-up meeting, studio sessions and Tribe leader meeting by every squad member is provided by the link below [Click to view](https://teams.microsoft.com/l/file/93331541-94C7-407F-8338-DA1BFE0E7A35?tenantId=d02378ec-1688-46d5-8540-1c28b5f470f6&fileType=docx&objectUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting%2FShared%20Documents%2FSurround%20AI%2FT2%202020%20(Squad%201)%2FMinutes%20of%20Meeting.docx&baseUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting&serviceName=teams&threadId=19:7e7092a286674cf99464a987cc2694bd@thread.tacv2&groupId=10567a03-e6be-45e6-b917-f7b8920e9c9b).

Appendix E:

* **Squad Presentation:** Panel presentation for the project can be viewed using the hyperlink [Click to view](https://teams.microsoft.com/l/file/18F47187-A050-4B43-BB69-6C7AC6CA1DC1?tenantId=d02378ec-1688-46d5-8540-1c28b5f470f6&fileType=pptx&objectUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting%2FShared%20Documents%2FSurround%20AI%2FT2%202020%20(Squad%201)%2FSurroundAI%20(Squad%201)%20-%20Panel%20Presentation.pptx&baseUrl=https%3A%2F%2Fdeakin365.sharepoint.com%2Fsites%2FData2IntelligenceConsulting&serviceName=teams&threadId=19:7e7092a286674cf99464a987cc2694bd@thread.tacv2&groupId=10567a03-e6be-45e6-b917-f7b8920e9c9b)